

02 135

RSU

日 本 国 特 許 庁

JAPAN PATENT OFFICE

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office

出 願 年 月 日

Date of Application:

2002年 7月31日

出 願 番 号

Application Number:

特願2002-222688

[ST.10/C]:

[JP2002-222688]

出 願 人

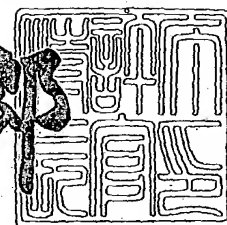
Applicant(s):

インターナショナル・ビジネス・マシーンズ・コーポレーション

2002年12月13日

特 許 庁 長 官
Commissioner,
Japan Patent Office

太田 信一郎



出証番号 出証特2002-3098278

【書類名】 特許願

【整理番号】 JP9020135

【あて先】 特許庁長官殿

【国際特許分類】 G06F 17/21
G06F 17/30

【発明者】

【住所又は居所】 神奈川県大和市下鶴間1623番地14 日本アイ・ピー・エム株式会社 大和事業所内

【氏名】 片岡 克尚

【特許出願人】

【識別番号】 390009531

【氏名又は名称】 インターナショナル・ビジネス・マシーンズ・コーポレーション

【代理人】

【識別番号】 100086243

【弁理士】

【氏名又は名称】 坂口 博

【代理人】

【識別番号】 100091568

【弁理士】

【氏名又は名称】 市位 嘉宏

【代理人】

【識別番号】 100108501

【弁理士】

【氏名又は名称】 上野 剛史

【復代理人】

【識別番号】 100085408

【弁理士】

【氏名又は名称】 山崎 隆

【手数料の表示】

【予納台帳番号】 117560

【納付金額】 21,000円

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【包括委任状番号】 9706050

【包括委任状番号】 9704733

【包括委任状番号】 0207860

【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 構造化文書用インターフェース装置、構造化文書用処理方法、及びプログラム

【特許請求の範囲】

【請求項 1】 アプリケーション・プログラムから処理要求のあった構造化文書について、字句解析の処理と、該構造化文書に係る一連のイベントを処理要求元のアプリケーション・プログラムへ順番に通知する処理を実施する構造化文書用インターフェース装置において、

アプリケーション・プログラムへ通知した一連のイベントをイベント・セット情報としてその構造化文書に対応付けてキャッシュにストアするストア手段、及び

処理対象の構造化文書についてそのイベント・セット情報が前記キャッシュに有るときは、該構造化文書に係る一連のイベントをキャッシュのイベント・セット情報に基づいて処理要求元のアプリケーション・プログラムへ順番に通知する第 1 の通知手段、

を有していることを特徴とする構造化文書用インターフェース装置。

【請求項 2】 そのイベント・セット情報がキャッシュに無い構造化文書については、それを読込んで、字句解析の処理と、該構造化文書に係る一連のイベントを処理要求元のアプリケーション・プログラムへ順番に通知する処理を実施する第 2 の通知手段を有し、

前記ストア手段は、第 2 の通知手段がアプリケーション・プログラムへ通知した前記一連のイベントをイベント・セット情報としてその構造化文書に対応付けてキャッシュにストアすることを特徴とする請求項 1 記載の構造化文書用インターフェース装置。

【請求項 3】 処理要求対象の構造化文書について、それを読込んで、字句解析の処理と、該構造化文書に係る一連のイベントを処理要求元のアプリケーション・プログラムへ順番に通知する処理と、さらに該通知した一連のイベントをイベント・セット情報としてその構造化文書に対応付けてキャッシュにストアする処理とを実施する第 1 の処理手段、

処理要求対象の構造化文書についてのキャッシュのイベント・セット情報を読み込んで、該イベント・セット情報に係る一連のイベントを順番に、処理要求元のアプリケーション・プログラムへ通知する第2の処理手段、及び

処理対象の構造化文書についてそのイベント・セット情報がキャッシュに有るか無いかを調べそのイベント・セット情報がキャッシュに無いときには前記第1の処理手段に処理対象の構造化文書についての処理を委譲し、また、そのイベント・セット情報がキャッシュに有るときには前記第2の処理手段に処理対象の構造化文書についての処理を委譲する制御手段、

を有していることを特徴とする構造化文書用インターフェース装置。

【請求項4】 前記制御手段は、処理対象の構造化文書についてそのイベント・セット情報がキャッシュに有るか無いかを判断するとき、処理対象の構造化文書と同一ファイル名の構造化文書に係るイベント・セット情報がキャッシュに有ったとしても、それが現在の構造化文書に対して更新前の構造化文書に係るイベント・セット情報であるならば、処理対象の構造化文書についてそのイベント・セット情報がキャッシュに無いとの判断を下すことを特徴とする請求項3記載の構造化文書用インターフェース装置。

【請求項5】 前記制御手段は、処理要求対象の構造化文書を処理要求元のアプリケーション・プログラムからURIにより通知されるパーサーであることを特徴とする請求項4記載の構造化文書用インターフェース装置。

【請求項6】 前記第1及び第2の処理手段はそれぞれ対応のパーサーに実装されるのに対し、前記制御手段は処理要求元のアプリケーション・プログラムに実装されていることを特徴とする請求項4記載の構造化文書用インターフェース装置。

【請求項7】 アプリケーション・プログラムから処理要求のあった構造化文書について、字句解析の処理と、該構造化文書に係る一連のイベントを処理要求元のアプリケーション・プログラムへ順番に通知する処理を実施する構造化文書用処理方法において、

アプリケーション・プログラムへ通知した一連のイベントをイベント・セット情報としてその構造化文書に対応付けてキャッシュにストアする第1のステップ

、及び

処理対象の構造化文書についてそのイベント・セット情報が前記キャッシュに有るときは、該構造化文書に係る一連のイベントをキャッシュのイベント・セット情報に基づいて処理要求元のアプリケーション・プログラムへ順番に通知する第2のステップ、

を有していることを特徴とする構造化文書用処理方法。

【請求項8】 そのイベント・セット情報がキャッシュに無い構造化文書については、それを読込んで、字句解析の処理と、該構造化文書に係る一連のイベントを処理要求元のアプリケーション・プログラムへ順番に通知する処理を実施する第3のステップ、及び

第3のステップにおいてアプリケーション・プログラムへ通知した前記一連のイベントをイベント・セット情報としてその構造化文書に対応付けてキャッシュにストアする第4のステップ、

を有していることを特徴とする請求項7記載の構造化文書用処理方法。

【請求項9】 処理要求対象の構造化文書について、それを読込んで、字句解析の処理と、該構造化文書に係る一連のイベントを処理要求元のアプリケーション・プログラムへ順番に通知する処理と、さらに該通知した一連のイベントをイベント・セット情報としてその構造化文書に対応付けてキャッシュにストアする処理とを実施する標準処理用ステップ、

処理要求対象の構造化文書についてのキャッシュのイベント・セット情報を読込んで、該イベント・セット情報に係る一連のイベントを順番に、処理要求元のアプリケーション・プログラムへ通知する短縮処理用ステップ、及び

処理対象の構造化文書についてそのイベント・セット情報がキャッシュに有るか無いかを調べそのイベント・セット情報がキャッシュに無いときには前記標準処理用ステップを実行するように制御し、また、そのイベント・セット情報がキャッシュに有るときには前記短縮処理用ステップを実行するように制御する制御ステップ、

を有していることを特徴とする構造化文書用処理方法。

【請求項10】 前記制御ステップは、処理対象の構造化文書についてその

イベント・セット情報がキャッシュに有るか無いかを判断するとき、処理対象の構造化文書と同一ファイル名の構造化文書に係るイベント・セット情報がキャッシュに有ったとしても、それが現在の構造化文書に対して更新前の構造化文書に係るイベント・セット情報であるならば、処理対象の構造化文書についてそのイベント・セット情報がキャッシュに無いとの判断を下すことを特徴とする請求項 9 記載の構造化文書用処理方法。

【請求項 11】 前記制御ステップは、処理要求対象の構造化文書を処理要求元のアプリケーション・プログラムから URI により通知されるパーサーにおいて実施されるものであることを特徴とする請求項 10 記載の構造化文書用処理方法。

【請求項 12】 前記第 1 及び第 2 の処理ステップは、それぞれ対応のパーサーにおいて実施されるものであるのに対し、前記制御ステップは、構造化文書についての処理要求を出すアプリケーション・プログラムにおいて実施されるものであることを特徴とする請求項 10 記載の構造化文書用処理方法。

【請求項 13】 アプリケーション・プログラムから処理要求のあった構造化文書について、字句解析の処理と、該構造化文書に係る一連のイベントを処理要求元のアプリケーション・プログラムへ順番に通知する処理を実施する構造化文書用処理プログラムにおいて、

アプリケーション・プログラムへ通知した一連のイベントをイベント・セット情報としてその構造化文書に対応付けてキャッシュにストアする第 1 のステップ、及び

処理対象の構造化文書についてそのイベント・セット情報が前記キャッシュに有るときは、該構造化文書に係る一連のイベントをキャッシュのイベント・セット情報に基づいて処理要求元のアプリケーション・プログラムへ順番に通知する第 2 のステップ、

をコンピュータに実行させるための構造化文書用処理プログラム。

【請求項 14】 そのイベント・セット情報がキャッシュに無い構造化文書については、それを読込んで、字句解析の処理と、該構造化文書に係る一連のイベントを処理要求元のアプリケーション・プログラムへ順番に通知する処理を実

施する第3のステップ、及び

第3のステップにおいてアプリケーション・プログラムへ通知した前記一連のイベントをイベント・セット情報としてその構造化文書に対応付けてキャッシュにストアする第4のステップ、
をコンピュータに実行させるための請求項13記載の構造化文書用処理プログラム。

【請求項15】 処理要求対象の構造化文書について、それを読込んで、字句解析の処理と、該構造化文書に係る一連のイベントを処理要求元のアプリケーション・プログラムへ順番に通知する処理と、さらに該通知した一連のイベントをイベント・セット情報としてその構造化文書に対応付けてキャッシュにストアする処理とを実施する標準処理用ステップ、

処理要求対象の構造化文書についてのキャッシュのイベント・セット情報を読込んで、該イベント・セット情報に係る一連のイベントを順番に、処理要求元のアプリケーション・プログラムへ通知する短縮処理用ステップ、及び

処理対象の構造化文書についてそのイベント・セット情報がキャッシュに有るか無いかを調べそのイベント・セット情報がキャッシュに無いときには前記標準処理用ステップを実行するように制御し、また、そのイベント・セット情報がキャッシュに有るときには前記短縮処理用ステップを実行するように制御する制御ステップ、

をコンピュータに実行させるための構造化文書用処理プログラム。

【請求項16】 前記制御ステップは、処理対象の構造化文書についてそのイベント・セット情報がキャッシュに有るか無いかを判断するとき、処理対象の構造化文書と同一ファイル名の構造化文書に係るイベント・セット情報がキャッシュに有ったとしても、それが現在の構造化文書に対して更新前の構造化文書に係るイベント・セット情報であるならば、処理対象の構造化文書についてそのイベント・セット情報がキャッシュに無いとの判断を下すことをコンピュータに実行させるための請求項15記載の構造化文書用処理プログラム。

【請求項17】 前記制御ステップは、処理要求対象の構造化文書を処理要求元のアプリケーション・プログラムからURIにより通知されるパーサーにお

いて実施されるものであることをコンピュータに実行させるための請求項16記載の構造化文書用処理プログラム。

【請求項18】 前記第1及び第2の処理ステップは、それぞれ対応のパーサーにおいて実施されるものであるのに対し、前記制御ステップは、構造化文書についての処理要求を出すアプリケーション・プログラムにおいて実施されるものであることをコンピュータに実行させるための請求項16記載の構造化文書用処理プログラム。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は、構造化文書用インターフェース装置、方法及びプログラムに関し、詳しくはXML (eXtensible Markup Language) 文書等の構造化文書について、アプリケーション・プログラムからの処理要求に応じて文字コード調整及び字句解析の処理と共に、該アプリケーション・プログラムへ、構造化文書に係る一連のイベントを通知する処理とを実施する構造化文書用インターフェース装置、方法及びプログラムに関するものである。

【0002】

【従来の技術】

近年、XML文書は、構成情報等の静的な情報を記述するために、使用される傾向が増大している。XML文書は高い可読性をもつので、アプリケーション・プログラムは、XML文書をインタープリタ形式で、つまり、必要のつど、SAXパーサー (SAX: Simple API for XML) を起動して、SAXパーサーによりXML文書を解析させ、解析結果としてのイベントをSAXパーサーより通知される形式で、該XML文書进行处理するのが望ましいことが多い。

【0003】

従来のSAXパーサーは、アプリケーション・プログラムからの処理要求を受けると、一律に、(a) 該当XML文書を読み込み、(b) 文字コード調整、(c) 字句解析、及び(d) 処理要求して来たアプリケーション・プログラムへの一

連のイベントの通知を行っている。

【0004】

【発明が解決しようとする課題】

本発明の目的は、構造化文書についてのアプリケーション・プログラムからの処理要求に対して該構造化文書に対応の一連のイベントを該アプリケーション・プログラムへ通知する構造化文書用インターフェース装置、方法、及びプログラムにおいて、処理速度を改善することである。

【0005】

【課題を解決するための手段】

本発明の構造化文書用インターフェース装置及び方法は、キャッシュを利用して、アプリケーション・プログラムへ通知した一連のイベントの情報としてのイベント・セット情報をキャッシュにストアする。そして、次に、前回のアプリケーション・プログラムと同一又は別のアプリケーション・プログラムから、イベント・セット情報がキャッシュにストアしてある構造化文書について、処理要求があった場合には、該構造化文書についてキャッシュにストアしてあるイベント・セット情報に基づいて、該構造化文書に係る一連のイベントを順番に、今回の処理要求元のアプリケーション・プログラムへ通知する。こうして、構造化文書用インターフェース装置が、一度、処理して、キャッシュにイベント・セット情報をストアしてそれが残っている構造化文書について、同一又は別のアプリケーション・プログラムが構造化文書用インターフェース装置へ再度、処理要求して来たときは、該アプリケーション・プログラムへのイベント通知処理のための処理量及び処理時間を大幅に低減できる。また、キャッシュにストアされるイベント・セット情報はアプリケーション・ユニークなものではないので、すなわち特定のアプリケーション・プログラムのみへのイベント通知しか適用できないものではないので、キャッシュにストアされるイベント・セット情報の活用が向上する。

【0006】

構造化文書とは、各内容が所定の要素に分類されている文書を含む。各要素は階層化されていてもよいとする。すなわち、本発明の構造化文書には、構造化及

び階層化された文書を含む。構造化及び階層化された文書として例えばXML文書がある。構造化文書における各要素の内容は典型的にはテキストで記述される。構造化文書とは、各内容が文書構造上のどの要素に該当するかを示しつつ、要素内容を記述した文書構造定義文書と把握することもできる。構造化文書は文書構造を例えばタグと要素名とにより定義している。

【0007】

本発明の構造化文書用インターフェース装置は、アプリケーション・プログラムから処理要求のあった構造化文書について、字句解析の処理と、該構造化文書に係る一連のイベントを処理要求元のアプリケーション・プログラムへ順番に通知する処理を実施する。該構造化文書用インターフェース装置は、アプリケーション・プログラムへ通知した一連のイベントをイベント・セット情報としてその構造化文書に対応付けてキャッシュにストアするストア手段、及び処理対象の構造化文書についてそのイベント・セット情報がキャッシュに有るときは、該構造化文書に係る一連のイベントをキャッシュのイベント・セット情報に基づいて処理要求元のアプリケーション・プログラムへ順番に通知する第1の通知手段、を有している。

【0008】

本発明の構造化文書用インターフェース装置は、処理要求対象の構造化文書について、それを読込んで、字句解析の処理と、該構造化文書に係る一連のイベントを処理要求元のアプリケーション・プログラムへ順番に通知する処理と、さらに該通知した一連のイベントをイベント・セット情報としてその構造化文書に対応付けてキャッシュにストアする処理とを実施する第1の処理手段、処理要求対象の構造化文書についてのキャッシュのイベント・セット情報を読込んで、該イベント・セット情報に係る一連のイベントを順番に、処理要求元のアプリケーション・プログラムへ通知する第2の処理手段、及び処理対象の構造化文書についてそのイベント・セット情報がキャッシュに有るか無いかを調べそのイベント・セット情報がキャッシュに無いときには第1の処理手段に処理対象の構造化文書についての処理を委譲し、また、そのイベント・セット情報がキャッシュに有るときには第2の処理手段に処理対象の構造化文書についての処理を委譲する制御

手段、を有している。

【0009】

本発明の構造化文書用処理方法は、アプリケーション・プログラムから処理要求のあった構造化文書について、字句解析の処理と、該構造化文書に係る一連のイベントを処理要求元のアプリケーション・プログラムへ順番に通知する処理を実施する。該構造化文書用処理方法は、アプリケーション・プログラムへ通知した一連のイベントをイベント・セット情報としてその構造化文書に対応付けてキャッシュにストアする第1のステップ、及び処理対象の構造化文書についてそのイベント・セット情報がキャッシュに有るときは、該構造化文書に係る一連のイベントをキャッシュのイベント・セット情報に基づいて処理要求元のアプリケーション・プログラムへ順番に通知する第2のステップ、を有している。

【0010】

本発明の構造化文書用処理方法は、処理要求対象の構造化文書について、それを読込んで、字句解析の処理と、該構造化文書に係る一連のイベントを処理要求元のアプリケーション・プログラムへ順番に通知する処理と、さらに該通知した一連のイベントをイベント・セット情報としてその構造化文書に対応付けてキャッシュにストアする処理とを実施する標準処理用ステップ、処理要求対象の構造化文書についてのキャッシュのイベント・セット情報を読込んで、該イベント・セット情報に係る一連のイベントを順番に、処理要求元のアプリケーション・プログラムへ通知する短縮処理用ステップ、及び処理対象の構造化文書についてそのイベント・セット情報がキャッシュに有るか無いかを調べそのイベント・セット情報がキャッシュに無いときには標準処理用ステップを実行するように制御し、また、そのイベント・セット情報がキャッシュに有るときには短縮処理用ステップを実行するように制御する制御ステップ、を有している。

【0011】

【発明の実施の形態】

図1は構造化文書入手のためにインターネット/イントラネット101が利用されるシステムの構成図である。クライアント101は、複数の構造化文書サーバ102へインターネット/イントラネット101を介して接続されている。

例えば、クライアント 1 1 は、構造化文書 1 2 を使用する後述のアプリケーション・プログラム 1 1 (図 2) と共に、後述の構造化文書用インターフェース装置 1 0、2 0、又は 2 5 を実装する。クライアント 1 0 1 は、今回の処理対象の構造化文書 1 2 (図 2) を管理する構造化文書サーバ 1 0 2 へ URL (Uniform Resource Locator) を指定して HTTP (HyperText Transfer Protocol) を使ってアクセスし、該所望の構造化文書 1 2 を入手する。なお、アプリケーション・プログラム及び構造化文書用インターフェース装置を実装するコンピュータが、自身のローカルのハード・ディスク装置内に処理対象の構造化文書を有していることもあり、そのような場合はパス名やファイル名等を利用して、構造化文書を入手する。

【 0 0 1 2 】

図 2 は本発明の第 1 の実施形態としての構造化文書用インターフェース装置 1 0 の機能ブロック図である。図 2 における S 1 ~ S 7 は後述の図 3 のステップ S 1 ~ S 7 に対応する。構造化文書用インターフェース装置 1 0 は、種々のアプリケーション・プログラム 1 1, 1 1, . . . から種々の構造化文書 1 2, 1 2, . . . についての処理要求を受け付ける。構造化文書 1 2 は、例えば、構造化文書用インターフェース装置 1 0 及びアプリケーション・プログラム 1 1 を実装するコンピュータ又は該コンピュータとは LAN 又はインターネットを介して接続されているサーバの記憶装置の所定フォルダ内に存在する。構造化文書 1 2 は、階層化された構造化文書であってもよく、階層化された構造化文書として例えば XML 文書がある。アプリケーション・プログラム 1 1, 1 1, . . . は、例えば、構造化文書 1 2 の内容に基づく音声応答を電話問い合わせに対して実施するアプリケーション・プログラムや、オペレータの指示に応じて構造化文書 1 2 の内容に基づく案内画面を作るアプリケーション・プログラムである。以下、説明の便宜上、処理要求対象の構造化文書 1 2 を「構造化文書 A」と呼び、構造化文書 A についての処理要求を構造化文書用インターフェース装置 1 0 へ先に出したアプリケーション・プログラム 1 1 を「アプリケーション・プログラム A」と呼び、構造化文書 A についての処理要求を構造化文書用インターフェース装置 1 0 へアプリケーション・プログラム A の後に出したアプリケーション・プログラム

11を「アプリケーション・プログラムB」と呼ぶことにする。典型的には、アプリケーション・プログラムBはアプリケーション・プログラムAと異なるものであるが、アプリケーション・プログラムBがアプリケーション・プログラムAである場合もある。構造化文書用インターフェース装置10は、ストア手段15及び第1の通知手段17を備え、キャッシュ16を適宜、利用する。

【0013】

構造化文書用インターフェース装置10は、アプリケーション・プログラムAからの構造化文書Aについての処理要求に対して(S1)、構造化文書Aを読み込んで、構造化文書Aの文字コード調整及び字句解析を実施しつつ(S2)、構造化文書Aに係る一連のイベントを、今回の処理要求元であるアプリケーション・プログラムAへ順番に通知する(S3)。なお、文字コード調整とは、構造化文書が日本語XML文書である場合、例えばShift JISからユニコードへの変換である。典型的なXMLパーサーはXML文書をユニコードへ変換して処理する。典型的な構造化文書用インターフェース装置は、構造化文書Aの文字コード調整及び字句解析の両方を実施するが、一部の構造化文書用インターフェース装置は、構造化文書Aの文字コード調整を省略して、直ちに構造化文書Aの字句解析を実施することがある。構造化文書用インターフェース装置10のストア手段15は、アプリケーション・プログラムAへ通知した一連のイベントをイベント・セット情報として構造化文書Aに対応付けてキャッシュ16にストアする(S4)。注意すべきは、字句解析においてエラーが見つかったときは、キャッシュ16へのイベント・セット情報のストアは中止されることである。したがって、構造化文書Aに字句解析上のエラーが1個でもあれば、構造化文書Aについてのキャッシュ16へのイベント・セット情報のストアは行われぬ。該エラーには例えば、要素名の開始のタグはあるものの、終わりのタグが無いと言うものがある。構造化文書が例えばXML文書である場合の字句解析上のエラーは、W3C(World Wide Web Consortium)のXML1.0(1.0は本願出願時のバージョン)の推薦(Recommendation)に定義されている通りであり、XML1.0の仕様書で規定しているXML規則に違反するものはすべて、字句解析上のエラーに含まれる。次に、アプリケーショ

ン・プログラムBが構造化文書用インターフェース装置10へ構造化文書Aについての処理要求を出したとする(S5)。構造化文書用インターフェース装置10の第1の通知手段17は、アプリケーション・プログラムAが構造化文書Aに対する処理要求を出したときの処理全体を繰り返すことなく、具体的には、構造化文書Aの再び読込んで、アプリケーション・プログラムAの文字コード調整及び字句解析を再び実施する処理はせずに、構造化文書Aについてキャッシュ16にストアされているイベント・セット情報を読み込み(S6)、読込んだイベント・セット情報に基づいて一連のイベントを検出し、該一連のイベントをアプリケーション・プログラムBへ順番に通知する(S7)。

【0014】

図3は図2の構造化文書用インターフェース装置10の処理手順のフローチャートである。構造化文書用インターフェース装置10はアプリケーション・プログラム11に対して構造化文書API(Application Program Interface)としての機能を果たす。S1では、アプリケーション・プログラムAから構造化文書Aについての処理要求を受け付ける。S2では、構造化文書Aを読み込む。アプリケーション・プログラムAからの構造化文書Aについての処理要求では、構造化文書Aはパス名(構造化文書Aがローカル・ファイルの場合)又はURL(構造化文書Aをネットワークを介して入手する場合)で特定されており、構造化文書Aの読み込みは、構造化文書用インターフェース装置10を実装するコンピュータが直接(構造化文書Aがパス名で特定されているとき)又はHTTP(構造化文書AがURLで特定されているとき)を介して行われる。S3では、アプリケーション・プログラムAへ該構造化文書Aに係る一連のイベントを順番に通知する。S4では、S3において通知した一連のイベントをイベント・セット情報として構造化文書Aに対応付けてキャッシュ16にストアする。S5では、アプリケーション・プログラムBから構造化文書Aについての処理要求を受け付ける。S6では、キャッシュ16におけるアプリケーション・プログラムAについてのイベント・セット情報を読み出す。S7では、読み出したイベント・セット情報に基づいて一連のイベントを検出し、各イベントを順番にアプリケーション・プログラムBへ通知する。

【 0 0 1 5 】

図4は本発明の第2の実施形態としての構造化文書用インターフェース装置20の機能ブロック図である。図4におけるS1～S13は後述の図5のステップS1～S13に対応する。構造化文書用インターフェース装置20について、構造化文書用インターフェース装置10との同一部分についての説明は省略し、構造化文書用インターフェース装置10との相違点について説明する。構造化文書用インターフェース装置20では、第2の通知手段22が構造化文書用インターフェース装置10に対して付加されている。ここで次の仮定を行う。該仮定では、キャッシュ16には、構造化文書Aのイベント・セット情報がストアされているものの、構造化文書Aとは別の構造化文書としての構造化文書Bのイベント・セット情報がまだストアされていない状態であるとし、さらに、この状態において、アプリケーション・プログラムBが構造化文書用インターフェース装置20に対して構造化文書Bの処理を要求したとする（S10）。第2の通知手段22は、構造化文書Bについて、それを読込んで、文字コード調整及び字句解析を実施した後（S11）、字句上のエラーが無ければ、アプリケーション・プログラムBへ構造化文書Bについての一連のイベントを順番に通知する（S12）。一方、ストア手段15は、構造化文書用インターフェース装置10（図2）が構造化文書Aについてのイベント・セット情報をキャッシュ16にストアしたときと同様に、アプリケーション・プログラムBへ通知した一連のイベントをイベント・セット情報として構造化文書Bに対応付けてキャッシュ16にストアする（S13）。

【 0 0 1 6 】

図5は図4の構造化文書用インターフェース装置20の処理手順のフローチャートである。図5において、S1～S4は図3のS1～S4と同一であるので、説明は省略し、S10～S13についてのみ説明する。S10～S13は、構造化文書Aに対するS1～S4の処理に対して、構造化文書Bに対する処理となっており、内容はS1～S4にそれぞれ対応している。S10では、アプリケーション・プログラムBから構造化文書Bについての処理要求を受け付ける。S11では、構造化文書Bを読込む。アプリケーション・プログラムBからの構造化文

書Bについての処理要求では、構造化文書Aの場合と同様に、構造化文書Bはパス名又はURLで特定されている。S12では、アプリケーション・プログラムBへ該構造化文書Bに係る一連のイベントを順番に通知する。S13では、S12において通知した一連のイベントをイベント・セット情報として構造化文書Bに対応付けてキャッシュ16にストアする。

【0017】

図6は本発明の第3の実施形態としての構造化文書用インターフェース装置25の機能ブロック図である。図6におけるS20～S29は後述の図7のS20～S29に対応する。構造化文書用インターフェース装置25について構造化文書用インターフェース装置20との相違点を説明する。構造化文書用インターフェース装置25は、第1の処理手段26、第2の処理手段27、及び制御手段28を有している。制御手段28は、アプリケーション・プログラムAから或る構造化文書についての処理要求を受ける（S20）、該或る構造化文書を構造化文書Aと呼ぶことにする。制御手段28は、該構造化文書Aについてのイベント・セット情報がキャッシュ16にストアされていないか調べ（S21）、無いと分かれば、第1の処理手段26へ処理の指示を出す（S22）。第1の処理手段26は、構造化文書Aを読込んで、文字コード調整及び字句解析を実施する（S22）。第1の処理手段26は、構造化文書Aについての字句解析の結果、エラー無しと判明すれば、構造化文書Aについての一連のイベントを順番にアプリケーション・プログラムAへ通知する（S23）。第1の処理手段26は、また、該通知した一連のイベントをイベント・セット情報としてその構造化文書に対応付けてキャッシュ16にストアする（S24）。その後、制御手段28は、アプリケーション・プログラムB（アプリケーション・プログラムBは、アプリケーション・プログラムAであることもあり得る。）から或る構造化文書についての処理要求を受け付ける（S20）。制御手段28は、アプリケーション・プログラムから処理要求のあった構造化文書についてのイベント・セット情報がキャッシュ16に有るか無いかを調べる（S21）。S20で受け付けた該或る構造化文書が、構造化文書Aとは異なる構造化文書Bであるときは、構造化文書Bについてのイベント・セット情報はキャッシュ16には無いので、制御手段28は第1

の処理手段 2 6 に処理を委譲する (d e l e g a t e)。S 2 0 で受け付けた該或る構造化文書が、構造化文書 A であるときは、構造化文書 A についてのイベント・セット情報はキャッシュ 1 6 に有るので、制御手段 2 8 は第 2 の処理手段 2 7 に処理を委譲する (S 2 8)。第 2 の処理手段 2 7 は、キャッシュ 1 6 から構造化文書 A に係るイベント・セット情報を読み込み (S 2 8)、該イベント・セット情報に基づいて、構造化文書 A に係る一連のイベントを順番にアプリケーション・プログラム B へ通知する (S 2 9)。

【 0 0 1 8 】

図 7 は図 6 の構造化文書用インターフェース装置 2 5 の処理手順のフローチャートである。S 2 0、S 2 2 ~ S 2 4 は図 3 の S 1 ~ S 4 に対応する。S 2 8、S 2 9 は図 3 の S 6、S 7 に対応する。S 2 0 では、或るアプリケーション・プログラムから或る構造化文書についての処理要求を受け付ける。S 2 1 では、該或る構造化文書に係るイベント・セット情報がキャッシュ 1 6 に有るか無いかを調べ、無ければ、S 2 2 へ進み、また、有れば、S 2 8 へ進む。S 2 2 では、処理要求のあった構造化文書を読み込み、文字コード調整及び字句解析等の処理を行う。字句上のエラーがなかったならば、S 2 3 では、S 2 0 において処理要求を出して来たアプリケーション・プログラムに、処理対象の構造化文書に係る一連のイベントを通知する。S 2 4 では、S 2 3 において通知した一連のイベントをイベント・セット情報としてキャッシュ 1 6 にストアする。S 2 8 では、今回、処理要求の出された構造化文書に係るイベント・セット情報をキャッシュ 1 6 から読み出す。S 2 9 では、S 2 8 に読み出したイベント・セット情報に基づいて一連のイベントを順番に、S 2 0 で処理要求を出して来たアプリケーション・プログラムへ通知する。S 2 8、S 2 9 のステップは、S 2 2 ~ S 2 4 のステップに対して、文字コード調整や字句解析の処理が省かれるので、アプリケーション・プログラムへの一連のイベントの通知を早めることができる。

【 0 0 1 9 】

図 6 及び図 7 についての今までの説明では、イベント・セット情報がキャッシュ 1 6 に有る構造化文書が、そのイベント・セット情報についてのキャッシュ 1 6 へのストア後、変更されていないことを前提にしている。キャッシュ 1 6 の容

量上の制約のために、新規なイベント・セット情報をストアするだけ残存量がキャッシュ16に無いときは、例えばストア日時の古いイベント・セット情報から順番に捨てられていく。或る構造化文書についてのイベント・セット情報がキャッシュ16に捨てられずに存在する期間は、限定されているかもしれないものの、該存在期間中にも、ソース（Source）の構造化文書が更新されることがあり、キャッシュ16にファイル名の同じ構造化文書についてのイベント・セット情報があるとしても、ソースの構造化文書の更新を反映していない場合が起こり得る。これに対処するため、キャッシュ16のイベント・セット情報の基になっている構造化文書が最新版のものか否かを調べ、最新版のものでなければ、該構造化文書についてキャッシュ16におけるイベント・セット情報は利用しないことにする。なお、利用できないと判明したイベント・セット情報はキャッシュ16から直ちに削除することが望ましい。図8は構造化文書の更新に対処してキャッシュのイベント・セット情報を利用する処理手順のフローチャートである。図8のフローチャートでは、図7のS21の判断処理に加えて、S34～S36が追加されている。S21では、今回、処理要求のあった構造化文書に係るイベント・セット情報にあるか否かを判定するが、この判定では、単に、今回、処理要求のあった構造化文書と同一ファイル名Xの構造化文書に係るイベント・セット情報にあるか否かを判定する。すなわち、該判定では、処理対象の構造化文書についての更新の有無は問わない。S21の判定がNOであれば、S22へ進むのに対し、S21の判定がYESであれば、S34へ進む。S34では、構造化文書原物（ソース構造化文書）についての作成日時D1を検出する。通常のファイル・システムでは、ファイル自体に対してファイル作成日時等の管理情報が所定の記憶エリアに記録されており、構造化文書原物についての作成日時D1は該記憶エリアへのアクセスにより検出される。S35では、キャッシュ16におけるファイル名Xについてのイベント・セット情報の基になっている構造化文書の作成日時D2を検出する。S36では、D1がD2以降の日時か否か、すなわちキャッシュ16におけるファイル名Xについてのイベント・セット情報の基になっている構造化文書が最新版のものであるか否かを判定し、該判定がYESであれば、S28へ進み、NOであれば、S22へ進む。なお、ここでは更新情報の

確認を日時を用いて行ったが、更新の有無を確認できるのであれば他の方法を用いることもできる。ソースの構造化文書がローカル・ファイルである場合の文書の作成日時はタイム・スタンプを検出することにより検出でき、また、ソースの構造化文書がHTTPにより入手するものである場合の文書の更新状態は、Last-Modifiedのヘダー付加に対するサーバからの応答（サーバ応答が304であれば、更新無しで、200であれば更新有り。）に基づいて検出できる。

【0020】

制御手段28は、(a) XMLパーサーのような構造化文書用パーサに実装される場合、及び(b) アプリケーション・プログラム内に実装される場合がある。アプリケーション・プログラムが、構造化文書用インターフェース装置25へURI (Uniform Resource Identifiers。後述のURLと区別。)で処理対象の構造化文書を指定するときは、構造化文書用パーサーは、URIを読むことのできるの、自ら、アプリケーション・プログラムから処理要求された構造化文書が何であるかをURIから解釈できる。したがって、ケース(a)には何ら支障はない。これに対して、Java (Sun Microsystems, Inc.の米国及びその他の国における商標)で記述された一部のアプリケーション・プログラムは、データをバイト・ストリームで構造化文書用パーサーへ渡すことがある。すなわち、アプリケーション・プログラムは、処理要求対象の構造化文書を開いて、この開いた構造化文書Xの内容をバイト・ストリームの形式で構造化文書用パーサーへ渡すことがある。バイト・ストリーム、すなわち構造化文書の内容自体からは更新日時情報等の管理情報は検出不能となるので、構造化文書用パーサーがアプリケーション・プログラムからデータをバイト・ストリームで受け取る場合には、制御手段28は、キャッシュ16内に処理対象の構造化文書についてのイベント・セット情報が有るか無いかを判定することができない。これに対処するためには、ケース(b)のように、制御手段28はアプリケーション・プログラムに実装され、制御手段28は、アプリケーション・プログラムが処理対象の構造化文書を開くのに先立って、該構造化文書についての作成日時等の必要情報を取得する。そして、該構造化文書Xに係るイベント・セット情報がキャッシュ16にあり、

かつ該イベント・セット情報が基にしている構造化文書の日時が構造化文書原物の日時と同じであることを調べ、その結果に基づいて、構造化文書用パーサーの第1の処理手段26及び第2の処理手段27のいずれに処理を移譲するかを決定することになる。

【0021】

【実施例】

図9は構造化文書用インターフェース装置としてのSAXパーサー34の構成図である。SAXパーサー34は、キャッシュ使用決定用パーサー35、キャッシュ・データ生成用パーサー36、キャッシュ・データ再生用パーサー37、及びキャッシュ・データ管理部38を有している。SAXパーサー34及びアプリケーション・プログラム41は、典型的には、LAN、イントラネット又はインターネットを介して相互に接続されているサーバ及びクライアントにそれぞれ実装される。しかしながら、SAXパーサー34及びアプリケーション・プログラム41は、同一のコンピュータに実装されている場合もある。アプリケーション・プログラム41は、所定のXMLファイル48をパス名及びURL等で指定して、該XMLファイル48についての処理をSAXパーサー34に要求する。SAXパーサー34のXMLデータ情報取得部45は、アプリケーション・プログラム41から処理要求のあったXMLファイル48を読み込む。なお、XMLデータ情報取得部45は、XMLファイル48が、SAXパーサー34の実装されているコンピュータ内にあるときは、すなわちローカル・ファイルであるときは、パス名の指定により読み込むが、XMLファイル48が、SAXパーサー34の実装されているコンピュータへインターネットやイントラネットを介して接続されているサーバに蓄積されているときは、httpやhttps (http secured) によりURLを指定して、読み込むことになる。

【0022】

下記はXMLファイル48の内容を例示したものである。

[XMLファイル48の内容例]

```
<?xml version="1.0" encoding="Shift_JIS"?>
```

```
<!DOCTYPE GUIDef SYSTEM "GUIDef.dtd">
```

<GUIDef>

<Parts name="DirTree" class="editor.DirTree" vp="FileAsString"/>

<Window name="Main" clear="true">

<Layout class="java.awt.BorderLayout"/>

<Group name="tab" parts="Tab" pos="Center">

<Ctrl parts="Label" pos="North">

<Init>プロジェクトの新規作成</Init>

</Ctrl>

</Group>

<Event action="Cleaner"/>

<Pr name="Title">プロジェクトの選択</Pr>

</Window>

</GUIDef>

【 0 0 2 3 】

一方、キャッシュ使用決定用パーサー 3 5 のキャッシュ・データ確認部 4 6 は、キャッシュ・データ管理部 3 8 を使ってキャッシュ内を SAX イベント・セット情報について検索する。キャッシュ・データ管理部 3 8 が管理するキャッシュ内には、キャッシュの容量内で幾つかのアプリケーション・プログラム 4 1 について SAX イベント・セット情報が蓄積可能になっている。新規の SAX イベント・セット情報をキャッシュにストアしようとするときに、キャッシュにストアする容量が残されていない時は、古い方の SAX イベント・セット情報から適当個数、適宜、削除して、該新規の SAX イベント・セット情報をストアする容量を確保し、そこへ該新規の SAX イベント・セット情報をストアすることになる。各 XML ファイル 4 8 の作成日時の情報は例えばタイム・スタンプとして記録されており、ソースの XML ファイル 4 8 のタイム・スタンプから該 XML ファイル 4 8 の作成日時として最後の更新日時を知ることができる。キャッシュ・データ確認部 4 6 は、アプリケーション・プログラム 4 1 からキャッシュ使用決定用パーサー 3 5 へ今回、処理要求のあった XML ファイル 4 8 と同一ファイル名の XML ファイル 4 8 に対応付けられている SAX イベント・セット情報を検索し

、該検索 SAX イベント・セット情報の日時を確認する。キャッシュ使用決定用パーサー 35 は、XML データ情報取得部 45 が読込んだ現在の XML ファイル 48 の作成日時とキャッシュ・データ確認部 46 が確認した SAX イベント・セット情報の日時とを照合して、キャッシュ内の SAX イベント・セット情報が現在の構造化文書の内容を基にした SAX イベント・セット情報であるか否かを調べる。(a) アプリケーション・プログラム 41 より今回、処理要求のあった XML ファイル 48 について、同一ファイル名の XML 文書についての SAX イベント・セット情報がキャッシュに無いとき、又は (b) アプリケーション・プログラム 41 より今回、処理要求のあった XML ファイル 48 について、同一ファイル名の XML 文書についての SAX イベント・セット情報がキャッシュに有るものの、その SAX イベント・セット情報について、それが基にしている XML ファイル 48 の作成日時が現在の XML ファイル 48 の作成日時より古いとき、キャッシュ使用決定用パーサー 35 は、キャッシュの SAX イベント・セット情報は利用不可とし、処理要求対象の XML ファイル 48 についての以降の処理をキャッシュ・データ生成用パーサー 36 へ委譲する。これに対して、(c) アプリケーション・プログラム 41 より今回、処理要求のあった XML ファイル 48 について、同一ファイル名の XML 文書についての SAX イベント・セット情報がキャッシュに有り、かつその SAX イベント・セット情報に対して、それが基にしている XML ファイル 48 の作成日時が現在の XML ファイル 48 の作成日時より古くないとき、すなわち XML ファイル 48 が更新されていないとき、キャッシュ使用決定用パーサー 35 は、キャッシュの SAX イベント・セット情報は利用可能とし、処理対象の XML ファイル 48 についての以降の処理をキャッシュ・データ再生用パーサー 37 へ委譲する。

【0024】

キャッシュ・データ生成用パーサー 36 は、XML データ取得部 53、文字コード調整部 54、字句解析部 55、SAX イベント生成部 56、及び SAX イベント記録部 58 を有している。キャッシュ・データ生成用パーサー 36 がキャッシュ使用決定用パーサー 35 から処理を委譲されると、XML データ取得部 53 は、処理対象の XML ファイル 48 を読込んで、XML ファイル 48 の XML デ

ータを取得する。文字コード調整部 5 4 は、XML データ取得部 5 3 が読込んだ XML データの文字コード（例：Shift JIS）をユニコード（Unicode）へ変換する。字句解析部 5 5 は、XML ファイル 4 8 のユニコードに基づいて処理対象の XML ファイル 4 8 の字句解析を行う。字句解析部 5 5 が、XML ファイル 4 8 の字句解析においてエラーがないと判断したら、SAX イベント生成部 5 6 は、XML ファイル 4 8 に係る一連のイベントを生成する処理に移り、SAX イベント・ハンドラ 5 7 を介して、今回、XML ファイル 4 8 の処理を要求して来たアプリケーション・プログラム 4 1 へ一連のイベントを順番に通知する。エラーがあるときは、アプリケーション・プログラム 4 1 へは、一連のイベントは通知することなく、エラーがある旨を通知する。図 1 0 は SAX イベント生成部 5 6 がアプリケーション・プログラム 4 1 へ通知する SAX イベントを示している。SAX イベント・ハンドラ 5 7 を介する SAX イベント生成部 5 6 からアプリケーション・プログラム 4 1 への一連のイベントの通知は、SAX イベント生成部 5 6 に実装される OOP（Object Oriented Program）のメソッド（Method）を介して行われる。或る種のメソッドには、1 個以上のパラメータ（引数）が付属し、パラメータには、XML ファイル 4 8 の要素名（例：GUIDef, Parts）、及び属性名（例：name, class, vp）＝属性値（例："Dirtree", "editor, Dirtree"）が含まれる。このようなメソッドを実行するために、SAX イベント生成部 5 6, 6 3 は、各種のインターフェースを実装する必要がある。SAX イベント生成部 5 6, 6 3 が実装すべきインターフェースの例は、例えば、クラス org.xml.sax.DocumentHandler に定義されており、次の通りである。

```
public abstract void setDocumentLocator (Locator locator);
public abstract void startDocument ()
public abstract void endDocument ()
public abstract void startElement (String name, AttributeList atts)
public abstract void endElement (String name)
public abstract void characters (char ch[], int start, int length)
public abstract void ignorableWhitespace (char ch[], int start, int le
```


ngth)

```
public abstract void processingInstruction (String target, String data
)
```

【0025】

SAXイベント記録部58は、SAXイベント生成部56がアプリケーション・プログラム41へ通知した一連のイベントをSAXイベント・セット情報としてキャッシュ・データ管理部38を介してキャッシュへ記録する。このSAXイベント・セット情報には、XMLファイル48の更新チェックのために、SAXイベント・セット情報が基にしているXMLファイル48の作成日時情報を含める必要がある。図11及び図12はキャッシュに記録するSAXイベント・セット情報の第1及び第2のパートである。SAXイベント・セット情報の第1のパートでは、各レコードにはメソッドとパラメータとが記述されている。パラメータは要素名と属性とを含む。メソッド名と要素名とはコード化されている。第2のパートでは、要素名とコードとの対応関係が記述されている。メソッドについてコードとメソッド名との関係は次の通りである。メソッドについてのコードとメソッド名との関係は、XMLファイル48に関係なく、定義されている。

startDocument : 1

startElement : 2

endElement : 3

characters : 4

endDocument : 5

【0026】

図12の第2のパートでは、テーブルの1番目のレコードには、SAXイベント・セット情報が基にしているXML文書の作成日時が記録される。テーブルの2番目以降のレコードには、XMLファイル48の要素についてコードとタグ（要素名）との関係が記録される。XMLファイル48の要素についてコードとタグ（要素名）との関係は、各XMLファイル48ごとに定義される。キャッシュ使用決定用パーサ35におけるXMLファイル48の前述の更新チェックは、現在のXMLファイル48の作成日時と図12の1番目のレコードの日時と照合

することにより行う。

【0027】

キャッシュ・データ再生用パーサー37はSAXイベント生成部63及びSAXイベント再生部64を有している。キャッシュ・データ再生用パーサー37がキャッシュ使用決定用パーサー35から処理対象のXMLファイル48についての処理を委譲されると、SAXイベント再生部64は、キャッシュ・データ管理部38に指示して、構造化文書におけるSAXイベント・セット情報を自分に送らせ、一連のSAXイベントを再生する（換言すると、例えば、一連のSAXイベントの各イベントをその順番に対応付けて検出する。）。SAXイベント再生部64がキャッシュ・データ管理部38から受けるSAXイベント・セット情報は図11及び図12のものである。SAXイベント再生部64は、図12におけるXMLファイル48の要素についてのコードとタグ（要素名）との関係に基づいて、図11の要素コードを要素名に変換して、図10のリストを再現し、SAXイベント生成部63へ渡す。SAXイベント生成部63は、これを基に、各メソッドを図11のリストの上から順番に実行し、一連のイベントの各々かSAXイベント・ハンドラ57を介してアプリケーション・プログラム41へ通知される。

【0028】

図13は図9のキャッシュ使用決定用パーサー35の処理手順のフローチャートである。S51では、アプリケーション・プログラム41が今回、処理要求したXMLファイル48に係るキャッシュ・データ（＝キャッシュにおけるSAXイベント・セット情報）があるか否か、すなわち、該XMLファイル48のファイル名に対応付けられているキャッシュ・データがあるか否かを判定し、判定がYESであれば、S52へ進み、NOであれば、S54へ進む。S51の判定は、XMLファイル48の更新状況は調べておらず、処理対象のXMLファイル48のファイル名と同一のファイル名のXMLファイルに係るキャッシュ・データがキャッシュに有るか否かのみを調べている。S52では、処理対象のXMLファイル48に係るキャッシュ・データを読み込み、すなわち図12のデータを読み込み、該データの日時、すなわちキャッシュにおけるSAXイベント・セット情報

の基にしたXMLファイル48の作成日時を検出する。S53では、キャッシュにおけるキャッシュデータが、現在のXMLファイル48の更新前のXML文書を基にしたデータか否かを調べる。S53における判定がYESであれば、すなわち、キャッシュにおけるSAXイベント・セット情報が基にしたXML文書が更新前のものであるときは、S54へ進む。S53における判定がNOであれば、すなわち、キャッシュにおけるSAXイベント・セット情報が基にしたXML文書が現在のXML文書であるときは、S55へ進む。処理対象のXMLファイル48の処理について、S54では、キャッシュ・データ生成用パーサー36へ委譲し、S55では、キャッシュ・データ再生用パーサー37へ委譲する。

【0029】

図14は図9のキャッシュ・データ生成用パーサー36の処理手順のフローチャートである。S60では、処理対象のXMLファイル48の文字データを読み込む。S61では、S60で読込んだ文字データをユニコードへ変換する。日本語XML文書の文字コードは例えばShift JISに準拠している。S62では、字句解析処理を行う。S63では、処理対象のXMLファイル48において要素名定義のタグを見つけ次第、該要素名に対応の1個のSAXイベントを生成する。生成されたSAXイベントは、対応のメソッドを使って、イベント・ハンドラを呼び出すことによりアプリケーション・プログラム41へ通知される。S65では、XMLファイル48の最後(EOF)に到達したか否かを判定し、該判定がYESであれば、この処理手順を終了し、NOであれば、S60へ戻り、次の文字データを読み込む。

【0030】

図15は図9のキャッシュ・データ再生用パーサー37の処理手順のフローチャートである。S68では、処理対象のXMLファイル48に係るSAXイベント・セット情報としてのキャッシュ・データの所定分をキャッシュから読み込む。S69では、S68で読込んだ所定分のキャッシュ・データに基づいてSAXイベントを生成する。生成されたSAXイベントは、対応のメソッドを使ってイベント・ハンドラを呼び出すことによりアプリケーション・プログラム41へ通知される。S70では、SAXイベント・セット情報(図12)の最後(EOF)に

到達したか否かを判定し、該判定が Y E S であれば、この処理手順を終了し、 N O であれば、 S 6 8 へ戻る。

【 0 0 3 1 】

まとめとして本発明の構成に関して以下の事項を開示する。

(1) : アプリケーション・プログラムから処理要求のあった構造化文書について、字句解析の処理と、該構造化文書に係る一連のイベントを処理要求元のアプリケーション・プログラムへ順番に通知する処理を実施する構造化文書用インターフェース装置において、

アプリケーション・プログラムへ通知した一連のイベントをイベント・セット情報としてその構造化文書に対応付けてキャッシュにストアするストア手段、及び

処理対象の構造化文書についてそのイベント・セット情報がキャッシュに有るときは、該構造化文書に係る一連のイベントをキャッシュのイベント・セット情報に基づいて処理要求元のアプリケーション・プログラムへ順番に通知する第 1 の通知手段、

を有していることを特徴とする構造化文書用インターフェース装置。

(2) : そのイベント・セット情報がキャッシュに無い構造化文書については、それを読込んで、字句解析の処理と、該構造化文書に係る一連のイベントを処理要求元のアプリケーション・プログラムへ順番に通知する処理を実施する第 2 の通知手段を有し、

ストア手段は、第 2 の通知手段がアプリケーション・プログラムへ通知した一連のイベントをイベント・セット情報としてその構造化文書に対応付けてキャッシュにストアすることを特徴とする(1)記載の構造化文書用インターフェース装置。

(3) : 処理要求対象の構造化文書について、それを読込んで、字句解析の処理と、該構造化文書に係る一連のイベントを処理要求元のアプリケーション・プログラムへ順番に通知する処理と、さらに該通知した一連のイベントをイベント・セット情報としてその構造化文書に対応付けてキャッシュにストアする処理とを実施する第 1 の処理手段、

処理要求対象の構造化文書についてのキャッシュのイベント・セット情報を読込んで、該イベント・セット情報に係る一連のイベントを順番に、処理要求元のアプリケーション・プログラムへ通知する第2の処理手段、及び

処理対象の構造化文書についてそのイベント・セット情報がキャッシュに有るか無いかを調べそのイベント・セット情報がキャッシュに無いときには第1の処理手段に処理対象の構造化文書についての処理を委譲し、また、そのイベント・セット情報がキャッシュに有るときには第2の処理手段に処理対象の構造化文書についての処理を委譲する制御手段、

を有していることを特徴とする構造化文書用インターフェース装置。

(4): 制御手段は、処理対象の構造化文書についてそのイベント・セット情報がキャッシュに有るか無いかを判断するとき、処理対象の構造化文書と同一ファイル名の構造化文書に係るイベント・セット情報がキャッシュに有ったとしても、それが現在の構造化文書に対して更新前の構造化文書に係るイベント・セット情報であるならば、処理対象の構造化文書についてそのイベント・セット情報がキャッシュに無いとの判断を下すことを特徴とする(3)記載の構造化文書用インターフェース装置。

(5): 制御手段は、処理要求対象の構造化文書を処理要求元のアプリケーション・プログラムからURI (Uniform Resource Identifiers)により通知されるパーサーであることを特徴とする(4)記載の構造化文書用インターフェース装置。

【 0 0 3 2 】

(6): 第1及び第2の処理手段はそれぞれ対応のパーサーに実装されるのに対し、制御手段は処理要求元のアプリケーション・プログラムに実装されていることを特徴とする(4)記載の構造化文書用インターフェース装置。

(7): アプリケーション・プログラムから処理要求のあった構造化文書について、字句解析の処理と、該構造化文書に係る一連のイベントを処理要求元のアプリケーション・プログラムへ順番に通知する処理を実施する構造化文書用処理方法において、

アプリケーション・プログラムへ通知した一連のイベントをイベント・セット情報としてその構造化文書に対応付けてキャッシュにストアする第1のステップ

、及び

処理対象の構造化文書についてそのイベント・セット情報がキャッシュに有るときは、該構造化文書に係る一連のイベントをキャッシュのイベント・セット情報に基づいて処理要求元のアプリケーション・プログラムへ順番に通知する第2のステップ、

を有していることを特徴とする構造化文書用処理方法。

(8)：そのイベント・セット情報がキャッシュに無い構造化文書については、それを読込んで、字句解析の処理と、該構造化文書に係る一連のイベントを処理要求元のアプリケーション・プログラムへ順番に通知する処理を実施する第3のステップ、及び

第3のステップにおいてアプリケーション・プログラムへ通知した一連のイベントをイベント・セット情報としてその構造化文書に対応付けてキャッシュにストアする第4のステップ、

を有していることを特徴とする(7)記載の構造化文書用処理方法。

(9)：処理要求対象の構造化文書について、それを読込んで、字句解析の処理と、該構造化文書に係る一連のイベントを処理要求元のアプリケーション・プログラムへ順番に通知する処理と、さらに該通知した一連のイベントをイベント・セット情報としてその構造化文書に対応付けてキャッシュにストアする処理とを実施する標準処理用ステップ、

処理要求対象の構造化文書についてのキャッシュのイベント・セット情報を読込んで、該イベント・セット情報に係る一連のイベントを順番に、処理要求元のアプリケーション・プログラムへ通知する短縮処理用ステップ、及び

処理対象の構造化文書についてそのイベント・セット情報がキャッシュに有るか無いかを調べそのイベント・セット情報がキャッシュに無いときには標準処理用ステップを実行するように制御し、また、そのイベント・セット情報がキャッシュに有るときには短縮処理用ステップを実行するように制御する制御ステップ、

を有していることを特徴とする構造化文書用処理方法。

(10)：制御ステップは、処理対象の構造化文書についてそのイベント・セット

情報がキャッシュに有るか無いかを判断するとき、処理対象の構造化文書と同一ファイル名の構造化文書に係るイベント・セット情報がキャッシュに有ったとしても、それが現在の構造化文書に対して更新前の構造化文書に係るイベント・セット情報であるならば、処理対象の構造化文書についてそのイベント・セット情報がキャッシュに無いとの判断を下すことを特徴とする(9)記載の構造化文書用処理方法。

【0033】

(11): 制御ステップは、処理要求対象の構造化文書を処理要求元のアプリケーション・プログラムからURIにより通知されるパーサーにおいて実施されるものであることを特徴とする(10)記載の構造化文書用処理方法。

(12): 第1及び第2の処理ステップは、それぞれ対応のパーサーにおいて実施されるものであるのに対し、制御ステップは、構造化文書についての処理要求を出すアプリケーション・プログラムにおいて実施されるものであることを特徴とする(10)記載の構造化文書用処理方法。

(13): アプリケーション・プログラムから処理要求のあった構造化文書について、字句解析の処理と、該構造化文書に係る一連のイベントを処理要求元のアプリケーション・プログラムへ順番に通知する処理を実施する構造化文書用処理プログラムにおいて、

アプリケーション・プログラムへ通知した一連のイベントをイベント・セット情報としてその構造化文書に対応付けてキャッシュにストアする第1のステップ、及び

処理対象の構造化文書についてそのイベント・セット情報がキャッシュに有るときは、該構造化文書に係る一連のイベントをキャッシュのイベント・セット情報に基づいて処理要求元のアプリケーション・プログラムへ順番に通知する第2のステップ、

をコンピュータに実行させるための構造化文書用処理プログラム。

(14): そのイベント・セット情報がキャッシュに無い構造化文書については、それを読込んで、字句解析の処理と、該構造化文書に係る一連のイベントを処理要求元のアプリケーション・プログラムへ順番に通知する処理を実施する第3の

ステップ、及び

第3のステップにおいてアプリケーション・プログラムへ通知した一連のイベントをイベント・セット情報としてその構造化文書に対応付けてキャッシュにストアする第4のステップ、

をコンピュータに実行させるための(13)記載の構造化文書用処理プログラム。

(15): 処理要求対象の構造化文書について、それを読込んで、字句解析の処理と、該構造化文書に係る一連のイベントを処理要求元のアプリケーション・プログラムへ順番に通知する処理と、さらに該通知した一連のイベントをイベント・セット情報としてその構造化文書に対応付けてキャッシュにストアする処理とを実施する標準処理用ステップ、

処理要求対象の構造化文書についてのキャッシュのイベント・セット情報を読込んで、該イベント・セット情報に係る一連のイベントを順番に、処理要求元のアプリケーション・プログラムへ通知する短縮処理用ステップ、及び

処理対象の構造化文書についてそのイベント・セット情報がキャッシュに有るか無いかを調べそのイベント・セット情報がキャッシュに無いときには標準処理用ステップを実行するように制御し、また、そのイベント・セット情報がキャッシュに有るときには短縮処理用ステップを実行するように制御する制御ステップ、

をコンピュータに実行させるための構造化文書用処理プログラム。

【0034】

(16): 制御ステップは、処理対象の構造化文書についてそのイベント・セット情報がキャッシュに有るか無いかを判断するとき、処理対象の構造化文書と同一ファイル名の構造化文書に係るイベント・セット情報がキャッシュに有ったとしても、それが現在の構造化文書に対して更新前の構造化文書に係るイベント・セット情報であるならば、処理対象の構造化文書についてそのイベント・セット情報がキャッシュに無いとの判断を下すことをコンピュータに実行させるための(15)記載の構造化文書用処理プログラム。

(17): 制御ステップは、処理要求対象の構造化文書を処理要求元のアプリケーション・プログラムからURIにより通知されるパーサーにおいて実施されるも

のであることをコンピュータに実行させるための(16)記載の構造化文書用処理プログラム。

(18): 第1及び第2の処理ステップは、それぞれ対応のパーサーにおいて実施されるものであるのに対し、制御ステップは、構造化文書についての処理要求を出すアプリケーション・プログラムにおいて実施されるものであることをコンピュータに実行させるための(16)記載の構造化文書用処理プログラム。

【0035】

【発明の効果】

本発明によれば、処理要求のあった構造化文書についての一連のイベントを通知するとともに、該通知した一連のイベントをイベント・セット情報としてキャッシュにストアする。そして、次に、同一構造化文書について処理要求を再度、受付けたときは、キャッシュにストアされているイベント・セット情報に基づいて一連のイベントを検出し、通知するようになっているので、構造化文書用インターフェース装置の処理時間を大幅に短縮できる。

【図面の簡単な説明】

【図1】

構造化文書入手のためにインターネット／イントラネットが利用されるシステムの構成図である。

【図2】

本発明の第1の実施形態としての構造化文書用インターフェース装置の機能ブロック図である。

【図3】

図2の構造化文書用インターフェース装置の処理手順のフローチャートである。

【図4】

本発明の第2の実施形態としての構造化文書用インターフェース装置の機能ブロック図である。

【図5】

図4の構造化文書用インターフェース装置の処理手順のフローチャートである。

【図6】

本発明の第3の実施形態としての構造化文書用インターフェース装置の機能ブロック図である。

【図7】

図6の構造化文書用インターフェース装置の処理手順のフローチャートである。

【図8】

構造化文書の更新に対処してキャッシュのイベント・セット情報を利用する処理手順のフローチャートである。

【図9】

構造化文書用インターフェース装置としてのSAXパーサーの構成図である。

【図10】

SAXイベント生成部がアプリケーション・プログラムへ通知するSAXイベントを示す図である。

【図11】

キャッシュに記録するSAXイベント・セット情報の第1のパートである。

【図12】

キャッシュに記録するSAXイベント・セット情報の第2のパートである。

【図13】

図9のキャッシュ使用決定用パーサーの処理手順のフローチャートである。

【図14】

図9のキャッシュ・データ生成用パーサーの処理手順のフローチャートである。

【図15】

図9のキャッシュ・データ再生用パーサーの処理手順のフローチャートである。

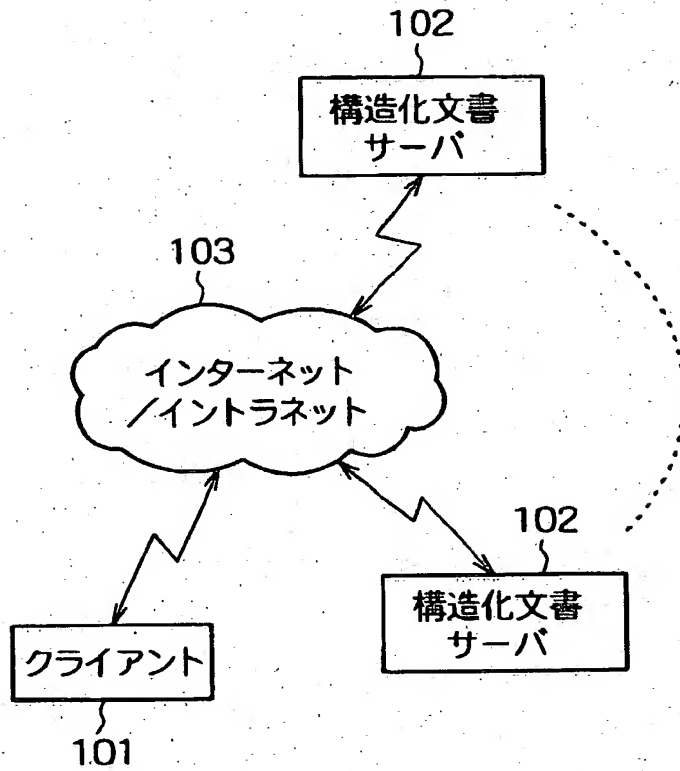
【符号の説明】

- 10 構造化文書用インターフェース装置
- 11 アプリケーション・プログラム
- 12 構造化文書
- 15 ストア手段
- 16 キャッシュ
- 17 第1の通知手段

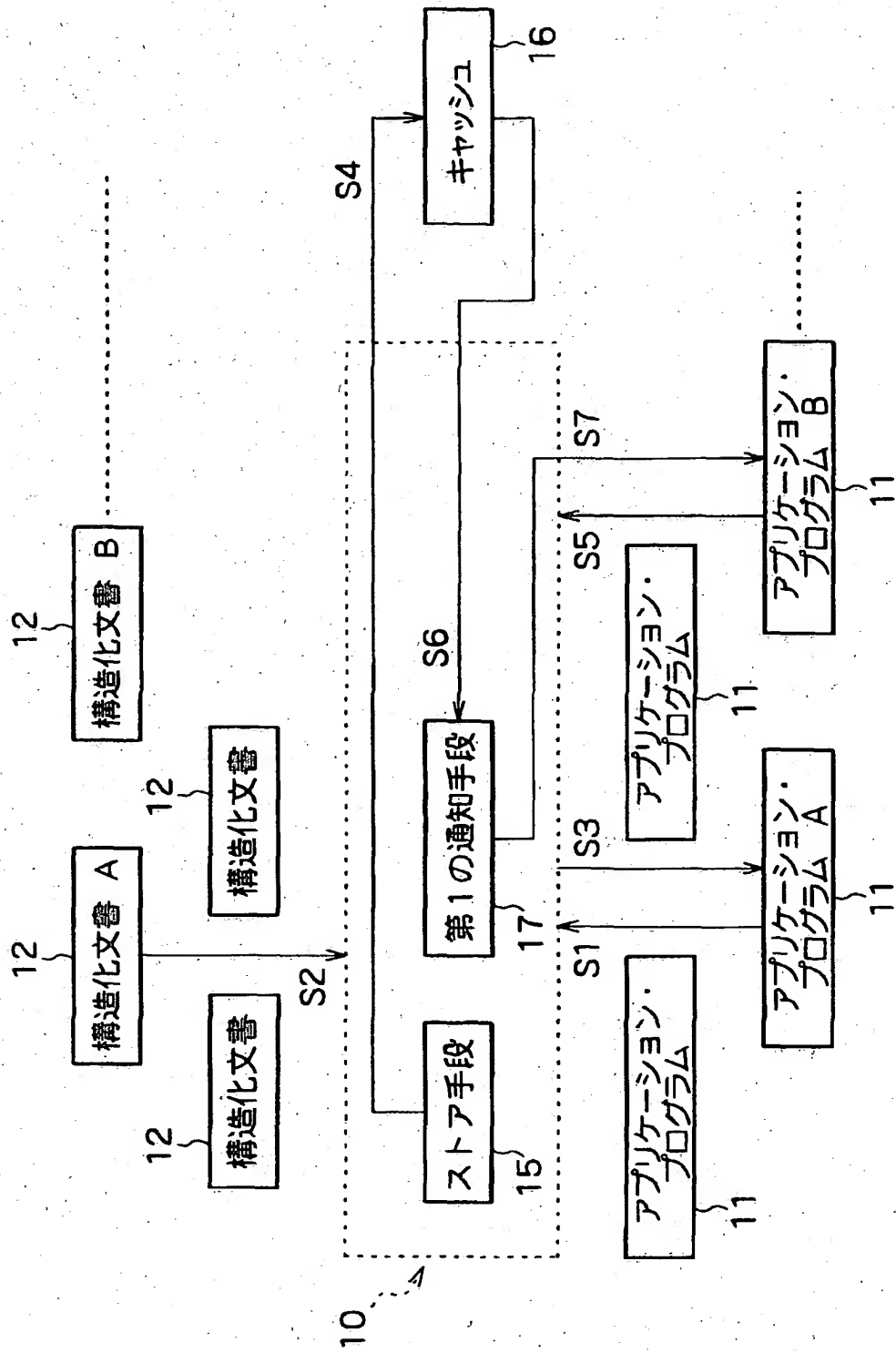
- 2 0 構造化文書用インターフェース装置
- 2 2 第 2 の通知手段
- 2 5 構造化文書用インターフェース装置
- 2 6 第 1 の処理手段
- 2 7 第 2 の処理手段
- 2 8 制御手段
- 3 5 キャッシュ使用決定用パーサー
- 3 6 キャッシュ・データ生成用パーサー
- 3 7 キャッシュ・データ再生用パーサー
- 4 1 アプリケーション・プログラム
- 4 8 XMLファイル

【書類名】 図面

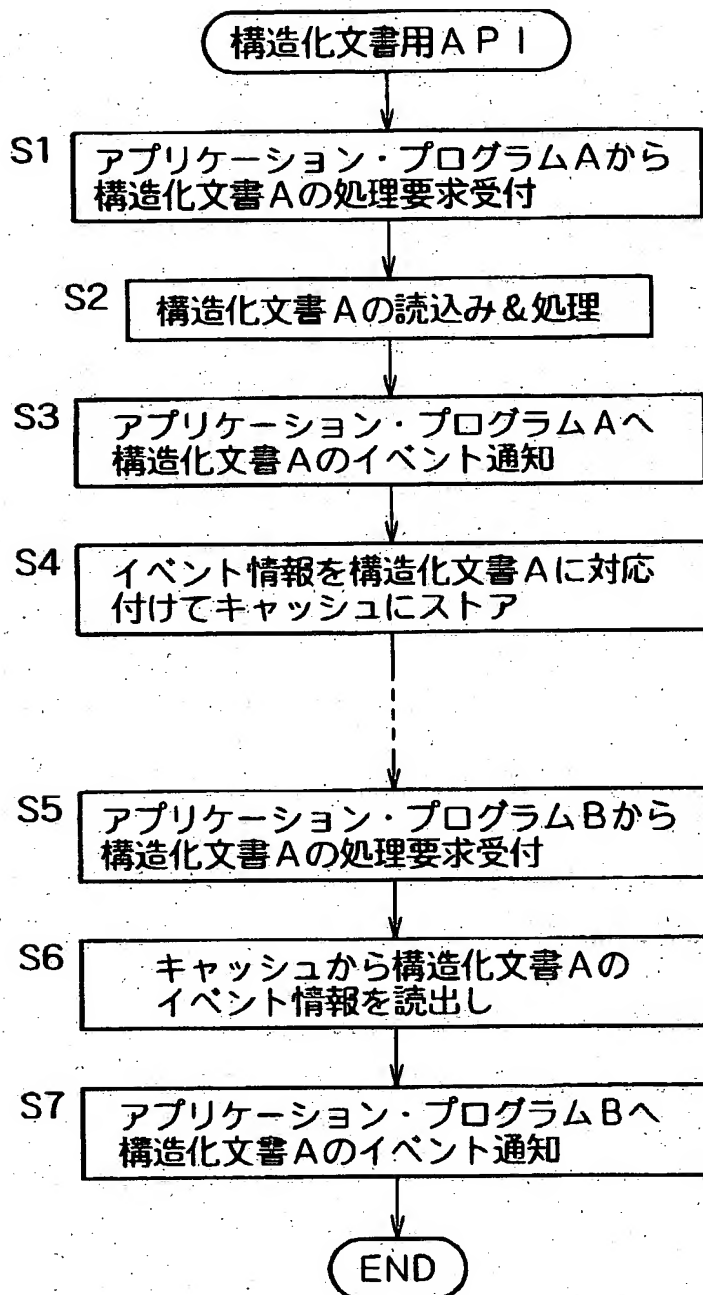
【図1】



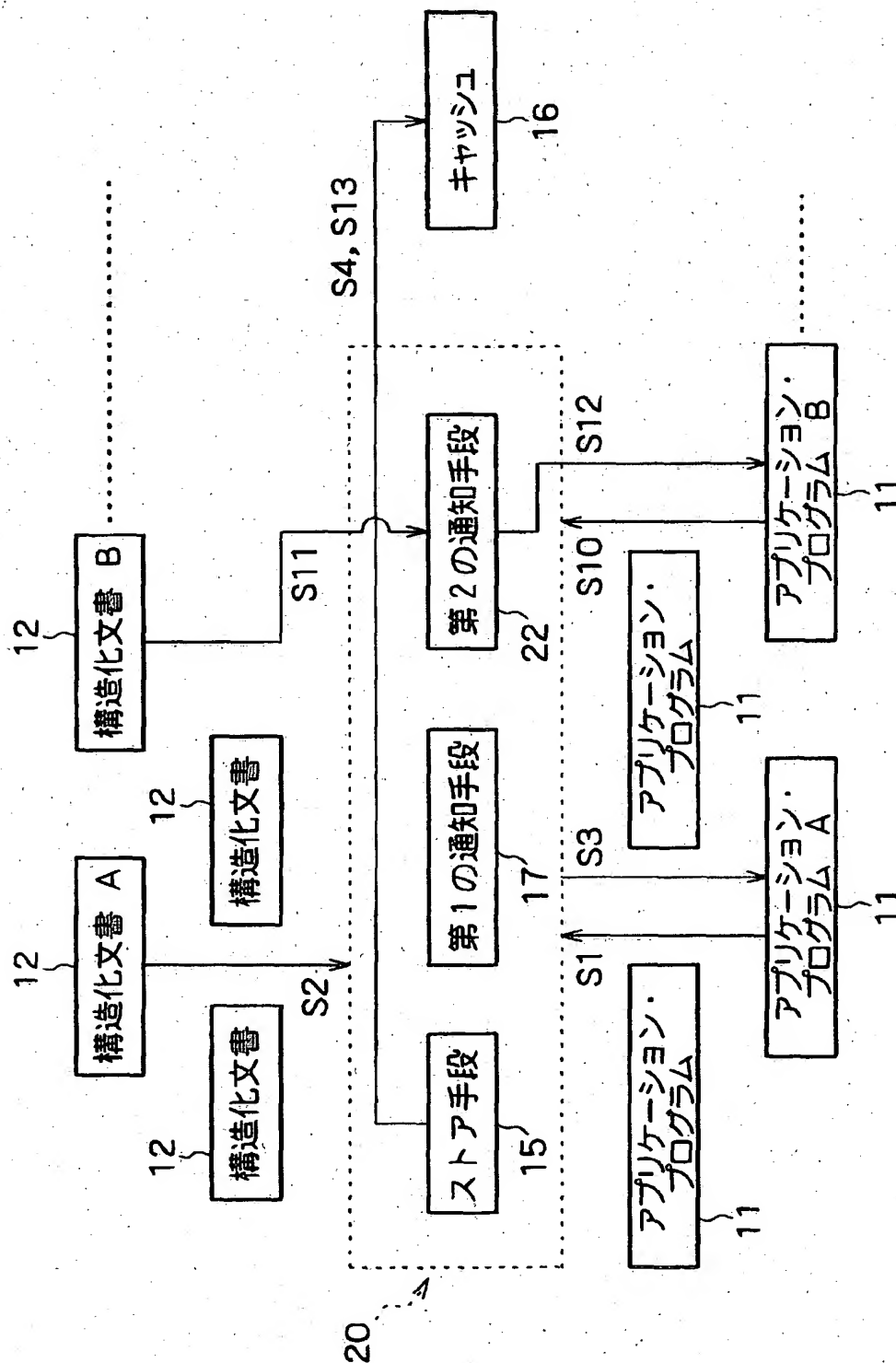
【図2】



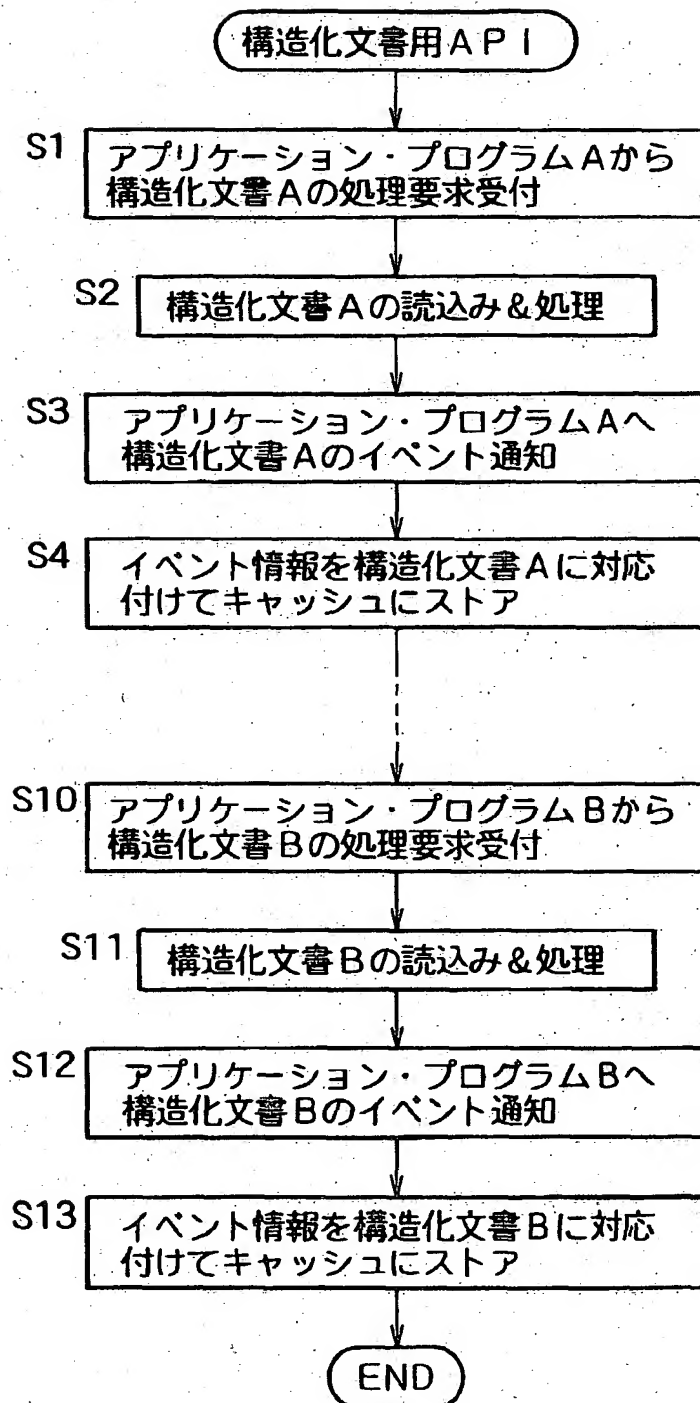
【図 3】



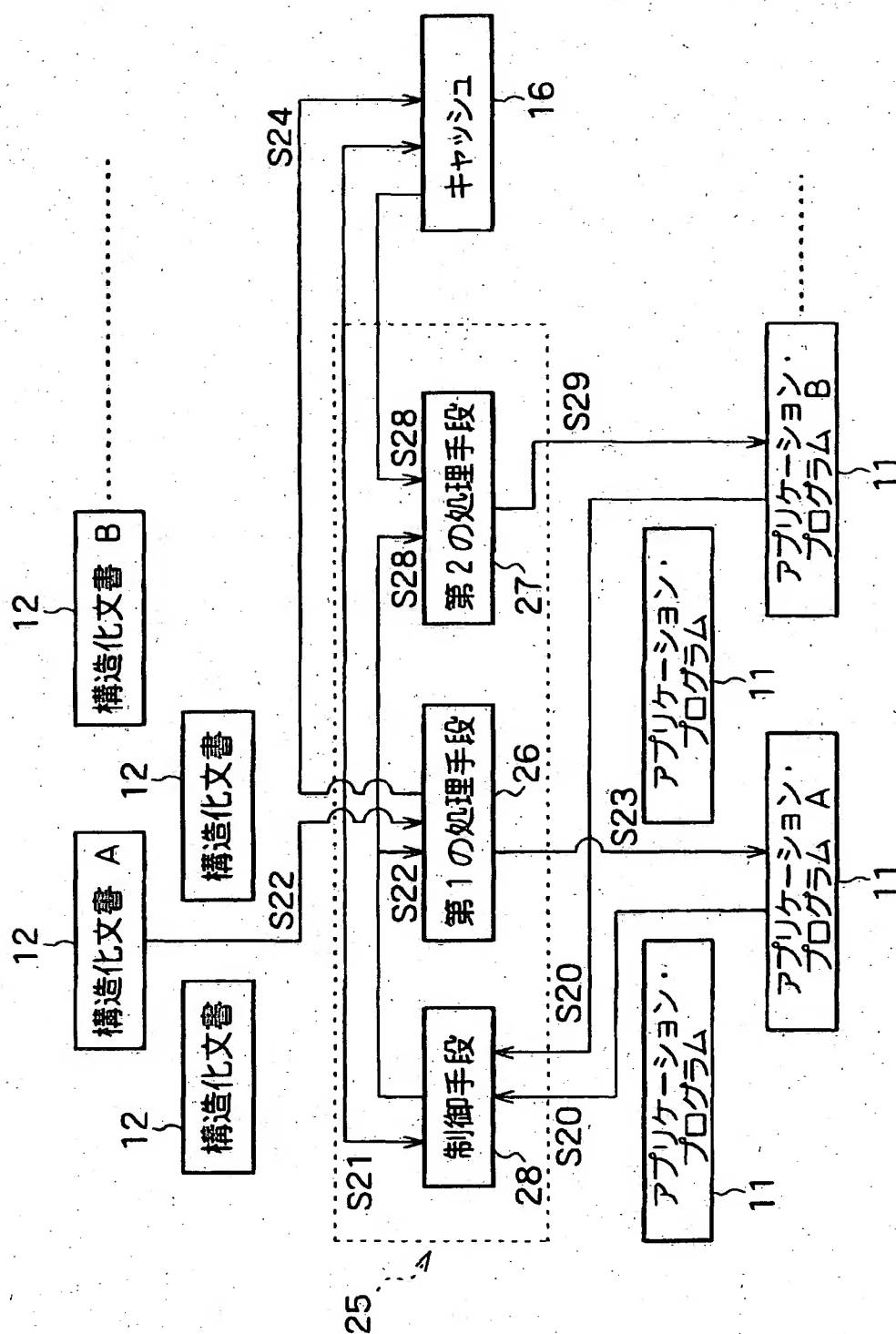
【图 4】



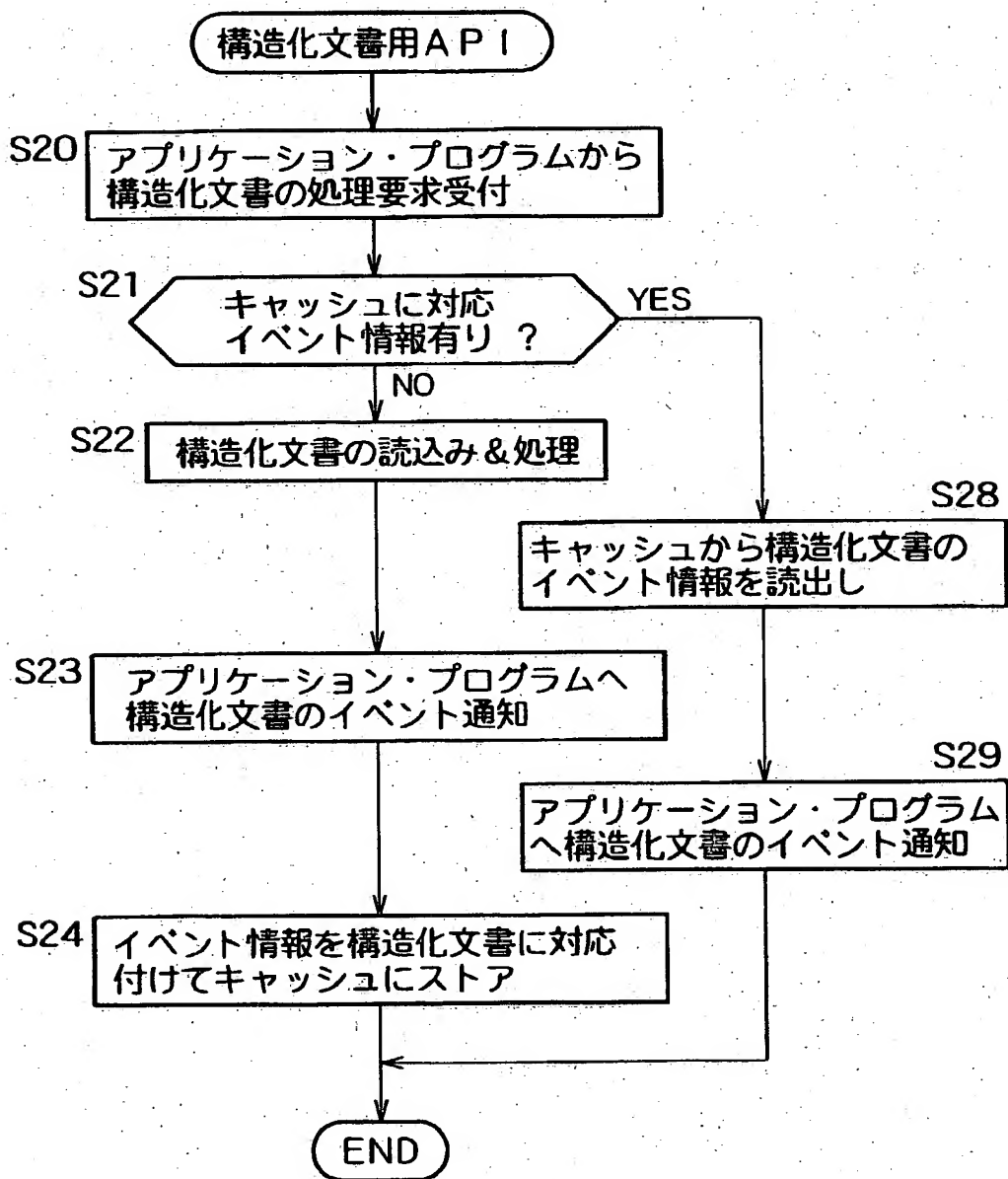
【図 5】



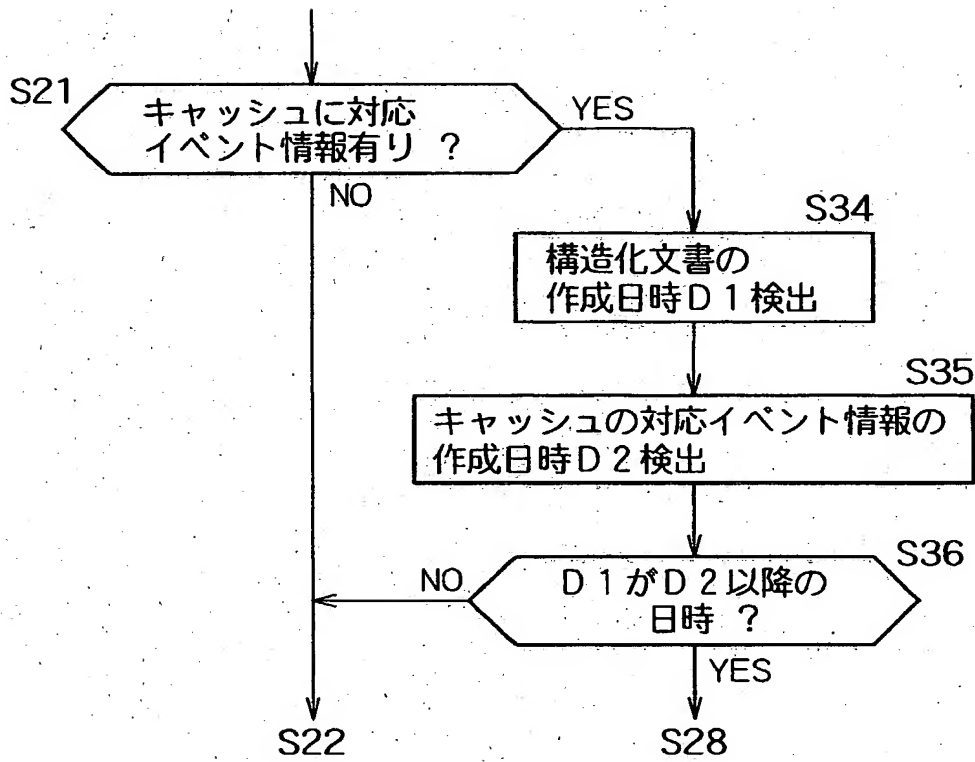
【図 6】



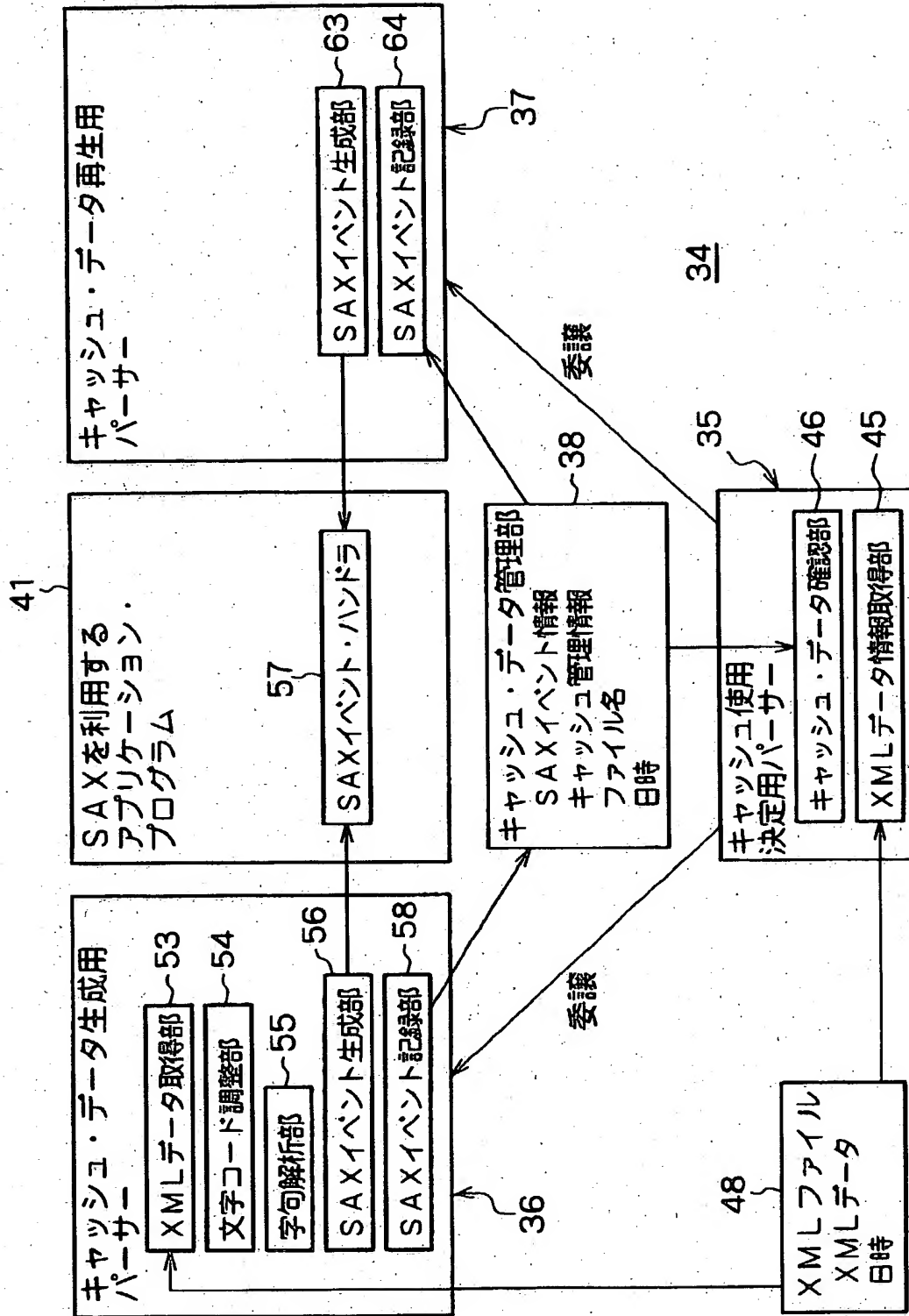
【図7】



【図 8】



【図9】



【図 10】

Method	パラメータ
startDocument	
startElement	GUIDef
startElement	Parts
endElement	Parts
startElement	Window
startElement	Layout
endElement	Layout
startElement	Group
startElement	Ctrl
startElement	Init
characters	プロジェクトの新規作成
endElement	Init
endElement	Ctrl
endElement	Group
startElement	Event
endElement	Event
startElement	Pr
characters	name="Title"
endElement	プロジェクトの選択
endElement	Pr
endElement	Window
endElement	GUIDef
endDocument	

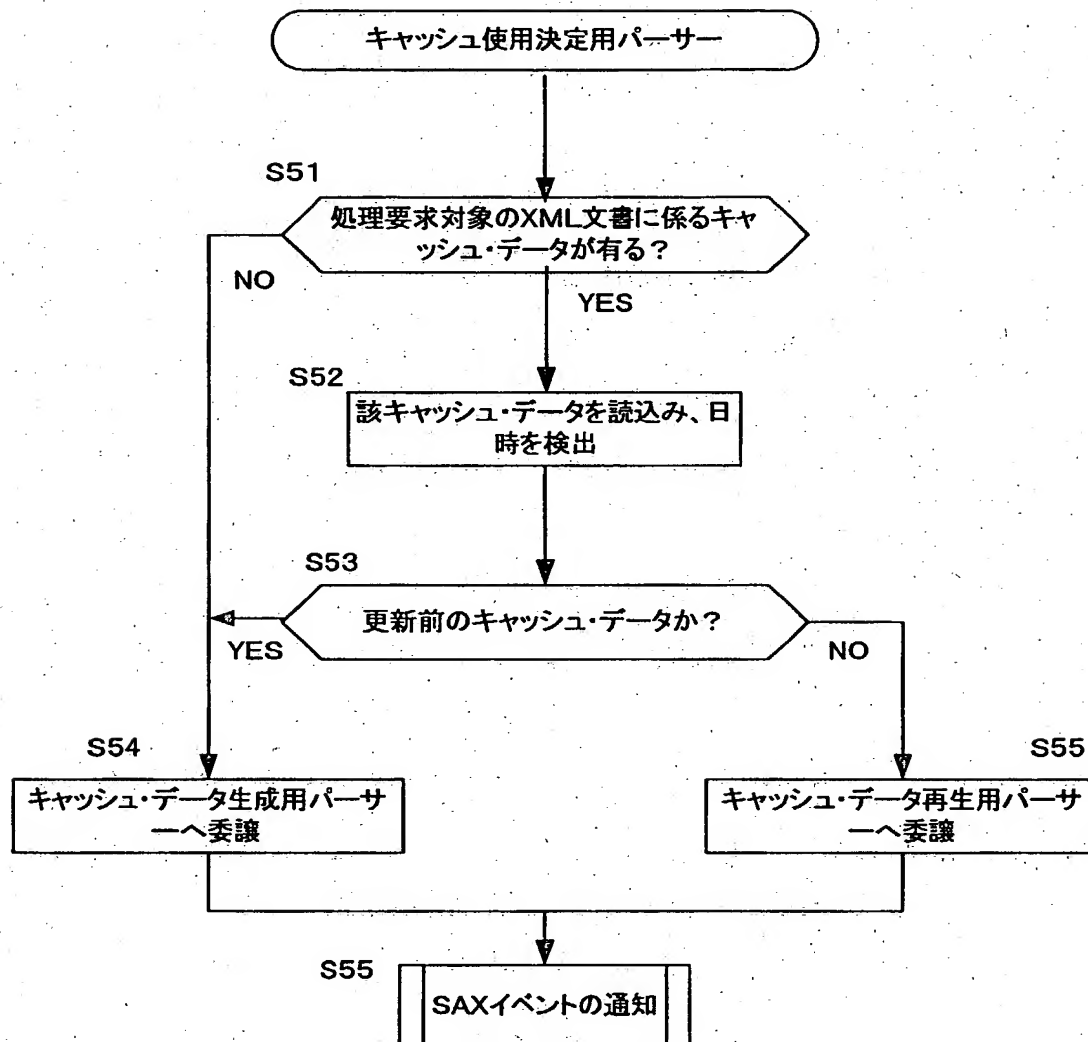
【図11】

Method	パラメータ
1	
2	1
2	2 name="DirTree", class="editor.DirTree", vp="FileAsString"
3	2
2	3 name="Main", clear="true"
2	4 class="java.awt.BorderLayout"
3	4
2	5 name="tab", parta="Tab", pos="Center"
2	6 parts="Label", pos="North"
2	7
4	プロジェクトの新規作成
3	7
3	6
3	5
2	8 action="Cleaner"
3	8
2	9 name="Title"
4	プロジェクトの選択
3	9
3	3
3	1
5	

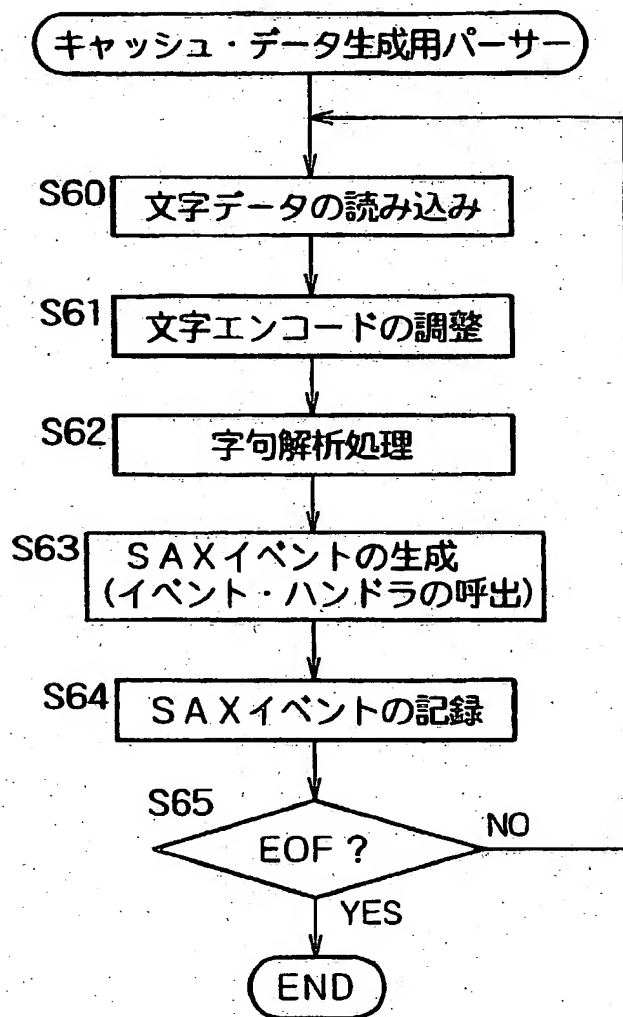
【図 12】

日時	xxxx/xx/xx xx:xx:xx.xxx
タグの文字テーブル	
1	GUIDef
2	Parts
3	Window
4	Layout
5	Group
6	Ctrl
7	Init
8	Event
9	Pr

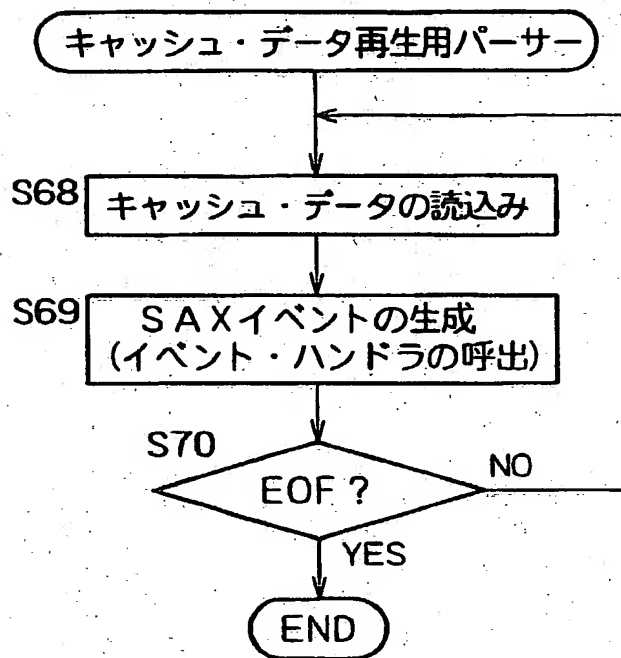
【図13】



【図14】



【図 1 5】



【書類名】 要約書

【要約】

【課題】 構造化文書12に係る一連のイベントをアプリケーション・プログラム11へ通知する処理を能率化する。

【解決手段】 構造化文書用インターフェース装置10は、アプリケーション・プログラム11からの構造化文書Aについての処理要求に対して、該構造化文書Aについて所定の処理を行ってから、構造化文書Aに係る一連のイベントを該アプリケーション・プログラム11へ順番に通知する。ストア手段15は、該通知した一連のイベントをイベント・セット情報としてキャッシュ16にストアする。次に、構造化文書Aについて同一又は別のアプリケーション・プログラム11から処理要求を受けたときには、キャッシュ16からイベント・セット情報を読み込んで、それに基づいて構造化文書Aに係る一連のイベントを再生し、該再生した一連のイベントを、構造化文書Aについての処理を要求して来たアプリケーション・プログラム11へ順番に通知する。

【選択図】 図2

認定 - 付加情報

特許出願の番号	特願2002-222688
受付番号	50201130160
書類名	特許願
担当官	土井 恵子 4264
作成日	平成14年 8月13日

<認定情報・付加情報>

【提出日】	平成14年 7月31日
【特許出願人】	
【識別番号】	390009531
【住所又は居所】	アメリカ合衆国10504、ニューヨーク州 アーモンク ニュー オーチャード ロード
【氏名又は名称】	インターナショナル・ビジネス・マシーンズ・コーポレーション
【代理人】	
【識別番号】	100086243
【住所又は居所】	神奈川県大和市下鶴間1623番地14 日本アイ・ビー・エム株式会社 大和事業所内
【氏名又は名称】	坂口 博
【代理人】	
【識別番号】	100091568
【住所又は居所】	神奈川県大和市下鶴間1623番地14 日本アイ・ビー・エム株式会社 大和事業所内
【氏名又は名称】	市位 嘉宏
【代理人】	
【識別番号】	100108501
【住所又は居所】	神奈川県大和市下鶴間1623番14 日本アイ・ビー・エム株式会社 知的所有権
【氏名又は名称】	上野 剛史
【復代理人】	申請人
【識別番号】	100085408
【住所又は居所】	東京都中央区日本橋2丁目1番1号 櫻正宗ビル 9階
【氏名又は名称】	山崎 隆

出 願 人 履 歴 情 報

識別番号 [390009531]

1. 変更年月日 2002年 6月 3日

[変更理由] 住所変更

住 所 アメリカ合衆国10504、ニューヨーク州 アーモンク ニ
ュー オーチャード ロード

氏 名 インターナショナル・ビジネス・マシーンズ・コーポレーショ
ン